Multi Agent System Analytics in Health Care

A Research Synopsis for the Pre-submission Défense towards partial Fulfilment of the Degree of Doctor of Philosophy(Ph.D)

in

Computer Science and Engineering

Submitted by

CHANDANITA THAKUR

Registration number: SVU/PhD/ENGG/CSE/001/2021

Swami Vivekananda University, Barrackpore, West Bengal, India

Under the Supervision of

Dr. Somsubhra Gupta (Supervisor)

Swami Vivekananda University, Barrackpore

And

Dr. Shibakali Gupta (Jt. Supervisor)

Department of Computer Science and Engineering, University Institute of Technology, Burdwan

MARCH, 2025

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SWAMI VIVEKANANDA UNIVERSITY

CHAPTER 1
INTRODUCTION

Evaluation of performance and efficiency is an important ingredient for any system design. A system can be comprised of many components. Components can be hardware or software. Each component needs technical support to have proper cooperation among them. Cooperation can be better provided if they have autonomy, reasoning, proactivity, mobility and capability of learning. These qualities are inherent for Agents. Software Agents can cooperate, coordinate and negotiate much better than the way we cooperate, coordinate and negotiate with each other. So it is obvious that the system designed with the cooperation of multiple Agents i.e. Multi Agent System can produce higher throughput than most of the other existing systems .To design a Multi Agent System it is very important to analyse the properties of Agents which is called semantics of Agent. To improve the performance of the Multi Agent System it is necessary to analyse and make a relationship between multiple Agent properties like proactivity, learning, reasoning, mobility, etc.

Health care is the most important sector which always seeks improvement. Multi Agent systems are considered as the best and most appropriate technology that can be applied to fulfil the need required in healthcare system. In Multi Agent system the agents are representing and acting on behalf of users and owners with very difficult goals and motivations. The agents cooperate, coordinate and negotiate with each other in the same way we cooperate, coordinate and negotiate with other people in day to day life and consequently can be used as the best technology to solve the problem in medical healthcare system.

It is very essential to develop a Multi Agent System to solve the issues that arise in Health care system. Heart disease is one of the most important imminence in human life. Early detection of Heart disease can prevent the death rate. Early detection depends on the analysis and prediction of the assortment of different parameters responsible for that disease. Health care industries have already accumulated enormous data. Scientists and researchers have paid their best attention to extract insights from the medical data by using Data Mining or Machine Learning. Different Machine Learning Techniques have helped Medical Science to achieve different levels of accuracy. There are different Machine Learning Algorithms to achieve accuracy but implementation of Multi Agent approach incorporated with Machine Learning Algorithm is an innovative approach in the field of Artificial Intelligence. Objective of the present paper is to apply a Multi Agent System with Machine Learning Algorithm and Execution of Machine Learning Algorithm using Intelligent Agent. The present paper concentrates on Twenty Six different machine learning Algorithms applied on Heart Disease dataset for flawless prediction of Heart Disease, its Accuracy, Balance Accuracy, ROC-AUC,F1-Score etc. The present paper attempts to predict heart disease using “Twenty Six " different Machine Learning Algorithms implemented by Agent and predicted Heart Disease with 93.8% accuracy. The paper also concentrates on development of the Multi Agent System and training it with the Best fit (Among Twenty-Six Machine Learning Algorithms) Algorithm. Agents in the newly created Multi Agent System have predicted the possibility of heart disease with an Accuracy of 96.8% approximately. The complete work has been implemented using Python (Jupyter Notebook). Implementation of Multi Agent System with Machine Learning Algorithm is the novelty of the present research work which can explore a new era in the Field of Artificial Intelligence. By systematically comparing the performance of these algorithms, it is identified that most effective models are required to refine and optimize prediction accuracy. So, further the final outcome of this research is the development of a hybrid Artificial Neural Network (ANN)-based algorithm, designed and implemented using

Python in a Jupyter Notebook environment. This hybrid ANN algorithm, leveraging the collaborative capabilities of intelligent agents, achieved a remarkable accuracy rate of 99.7%. The results underscore the transformative potential of multi-agent systems in medical diagnostics, paving the way for more accurate and reliable heart disease prediction models.

CHAPTER 2
ORGANIZATION OF THE THESIS

Details of the Chapters

Chapter 1 : This chapter elaborates background or motivation for study, Problem analysis, Specific contribution, Significance of the study etc.

Chapter 2: Provides the Organization of the Thesis.

Chapter 3: Provides a detailed survey about various related work on Multi Agent System applied on Health care

Chapter 4: Deals with analysing and making a relationship between multiple Agent properties like proactivity, learning, reasoning, mobility, etc. and finally proposing an efficient algorithm to calculate the efficiency of Agents based on their properties and make them to work simultaneously in a Multi Agent environment, for satisfying several goals.

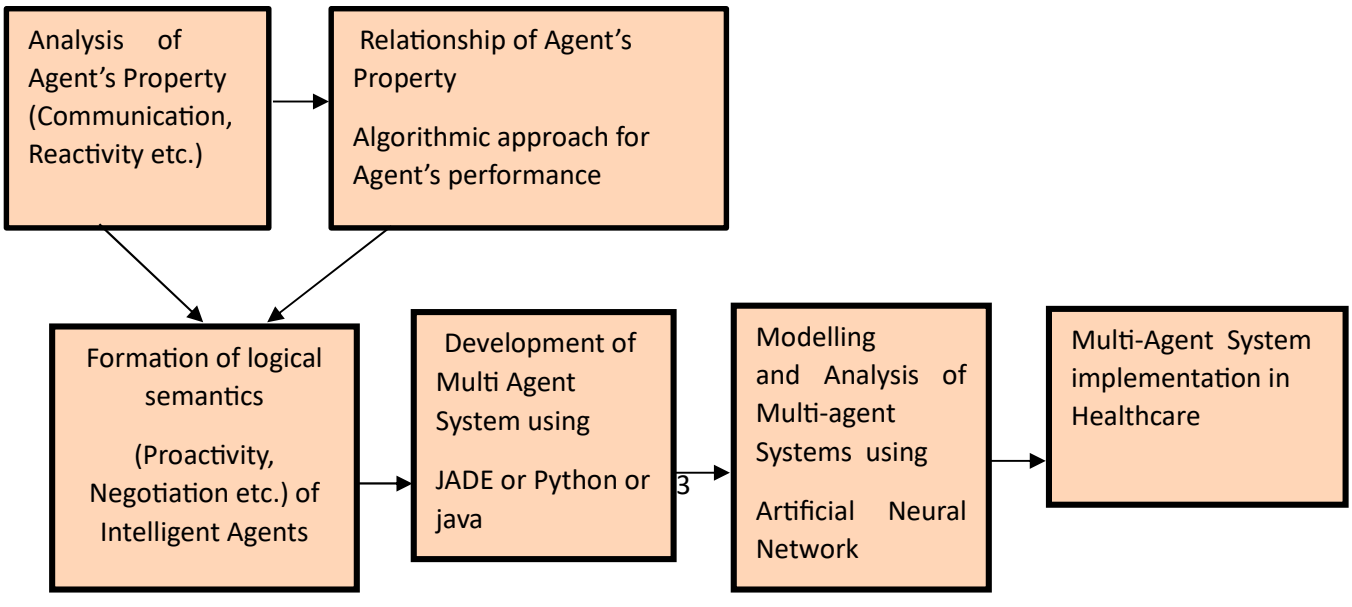
Chapter 5: Concentrates on development of the Multi Agent System and training it with Twenty-Six Machine Learning Algorithms. Agents in the newly created Multi Agent System have predicted the possibility of heart disease with an Accuracy of 96.8% approximately by using the best fit algorithm among all Twenty-six algorithms. Twenty-Six different machine learning Algorithms applied on heart disease dataset (a mixture of five different standard dataset) for flawless prediction of Heart Disease, its Accuracy, Balance Accuracy, ROC-AUC, F1-Score etc.

Chapter-6: Compares the performance of different Machine Learning Algorithms, it is identified that most effective models are required to refine and optimize prediction accuracy. So, further the final outcome of this research is the development of a hybrid Artificial Neural Network (ANN)-based algorithm, designed and implemented using Python in a Jupyter Notebook environment.

Chapter 7: Is used to design a Multi Agent based simulation model to show the spread of coronavirus based on Euclidean distance between every pair of Agents. Simulation of Multi Agent System is shown with the help of image viewer. Multi Agent Simulation is implemented using Jupyter notebook.

Chapter 8: Concludes the observations of the thesis and the future enhancements of the examined work.

Road Map of the present work


Analysis of Agent's Property (Communication, Reactivity etc.)
Relationship of Agent's Property
Algorithmic approach for Agent's performance
Formation of logical semantics
(Proactivity, Negotiation etc.) of Intelligent Agents
Development of Multi Agent System using
JADE or Python or java
Modelling and Analysis of Multi-agent Systems using
Artificial Neural Network
Multi-Agent System implementation in Healthcare

CHAPTER 3
LITERATURE SURVEY

The purpose of this chapter is to provide thorough review of application of Multi Agent System from 1960 to 2020 in Medical Domain and to guide future researchers to carry on their work in the same field.

Katia P. Sycara [1] has showed a Multi Agent Infrastructure called "RETSINA" which follows "BDI" ("Belief, Desire, and Intension") and follows the semantics like planning, scheduling, data collecting, coordinating etc.

Katia P. Sycara and **Michael Wooldridge** [2] have tried to find out how autonomous agent can interact with each other and where they can be applied etc.

Rahul Singh, and Katia Sycara [3] has proposed the need of interface of service provider and Agent Communication language (ACL). Currently Multi Agent System uses KQML, FIPA , emerging web service communication language like SOAP etc

Regina Sousa et al. [4] emphasized on the creation of "AIDAMonit", a surveillance platform developed and tested by ALGORITMI Center researchers, which is extremely beneficial for those who are currently using AIDS platform.

Maroua Ahmid [5][6]et al.proposes an cloud IoT system based on agents for remote monitoring of cardiovascular system which can collect, retrieve, store and analyze heart rate of patient.

Abu Bakar et al.[7] proposes an interrelated decision-making model (IDM) for Intelligent Decision Support System (IDSS) in healthcare that aims to have an effective decision by utilizing knowledge from previous and following treatment stages known as IDM-IDSS-healthcare which uses Multi Agent System.

Cardiac disease has been detected by using KNN, DT, NB,LR, RF [8] algorithm with 96.6% of accuracy based on the Cleveland Dataset. Cardiovascular disease has been diagnosed on Kaggle dataset with 87.28 percent of accuracy [9].Authors have used multiple algorithms i.e. RF,DT,MP,XGB.Heart disease diagnosis with LR, SVM, DT [10] machine learning algorithm is carried out with UCI dataset and achieved accuracy of 96.6 percent. By analyzing Cleveland Dataset, Heart disease is analyzed with 86.7 % accuracy [11] (by DT,GB,LR,SVM,RF algorithms).With accuracy 93.44%, Heart disease is predicted by using RF,KNN,LR,NB,GB,daB algorithms[12].Authors have executed the algorithms based on Cleveland Dataset. Next work is based on LR,NN,SVM,DT,RF algorithms and Cleveland Dataset, where detected accuracy is 82.18%[13].Heart disease is also predicted by LR,NN,SVM,DT,RF [14] algorithm with 90% accuracy where Data have been collected from hospitals, diagnostic centers and clinical centers in Bangladesh. Heart disease has been predicted with 86.7 percent [15] accuracy based on Real Time data. Heart Attack is detected with 94 % accuracy on UCL dataset and LR Machine Learning algorithm has been used [16].NB,DT ,WARM algorithm has been used [17] with 96.5% accuracy .Heart disease has been predicted by using Machine Learning algorithms with 91.67% accuracy [18].Cardiovascular disease has been predicted with 90% accuracy [19] (by applying SVM algorithm).Heart failure is detected with 87.03% accuracy by using LR,KNN,MNB,ET,RF,DT,GBC,DT,LGBM, HGBC,and

XGB algorithm [20]. Heart disease is diagnosed by LR,RF with 83% possibility [21].Heart disease is predicted with LR,SVM,KNN algorithms [22] with a maximum of 92.1% accuracy. With the help of LR, SVM, and KNN algorithms, Heart disease is detected with 93.44% of accuracy[23]. Cardiovascular disease prediction is done using the NB,KNN,LR algorithm with 87.5 % accuracy. Using UCI dataset [24].Heart disease has been diagnosed using KNN,RF algorithm [25] (with a maximum accuracy of 86.88%).Cardiovascular disease has been detected by using SGD [26] (SHDML framework).

A spatiotemporal COVID-19[27] transmission model was proposed based on creating an artificial agent based society. Connection was established between different agents and spreading of coronavirus was demonstrated with transformation rules, and social relations etc..To address [28] the COVID-19 issues at Nanjing "Lukou Airport", authors have simulated Passenger's movement and come up with an algorithm to generate the moving traces. "Mobile contact networks" have been developed with "a static network of staff and passengers". Every individual have been facilitated with complex dynamical network. Agent based [29] simulation model is designed to predict the impact of corona virus on human health. Demonstration of use of different brands of vaccines, respirators, types of masks are implemented using NetLogo in SEIR Framework. A data [30] driven Predictive model is designed to improve COVID-19 cases to improve public policy making. It helped to find the impact of coronavirus on global health and economy .AEDM model [31] is designed based on ambiguous set theory and useful in processing ambiguous pixel information. The proposed model has highlighted the edges of CT scan into three categories low ,moderate and severe infection. By using [32] a Mathematical Model called ARMA, autoregressive moving average model the authors have discussed the effect of corona on the Indian economy, society, and financial sector. A Machine [33] Learning based (KNN) model is proposed for quick identification and prediction of risk of corona so that early prediction can reduce the risk of corona. Prediction accuracy of the algorithm is "98.34%, Recall is 97% and an F1-Score is 0.97".Agent based [34] Mathematical simulation model was developed to show the evolution and spread of Corona virus .COVID-19 [35]Agent-based Simulator is developed to predict the trends of epidemic and simulate the resource need. This model also have supported the non-pharmaceutical need, hygiene measure, physical distancing etc. Covasim, [16] "an agent-based model (ABM) of COVID-19", was developed to measure the probability of epidemic spread based on "the region's demographic and geographical features". Data collection was made on the basis of number of people tested, diagnoses, critical cases, hospitalization etc. and those data was analysed region wise.

GAP ANALYSIS

From the analysis of Literature Survey it is very clear that Multi Agent System is applied in different types of Health care sector. Overall Findings from the review fifty six papers are

i. Most of the works have used Multi Agent System for Modelling Healthcare system.

ii. A good number of works have only Simulated the Health care System using Multiple Agents.

iii. Many works have not been implemented using Programming Language.

iv. Few of the implemented works have used JADE programming.

v. Implementation of Multi Agent System using Python and Machine Learning is rare in the literature between 2017 to 2021.

vi. Recent papers have implemented Machine Learning algorithms using python in Health care but Multi Agent systems implemented using Machine Learning algorithm is missing in Literature.

vii. Artificial Neural Network approach using Machine Learning algorithm for development and analysis of Multi Agent systems is missing in Literature.

viii. Less number of algorithms are used for prediction of any disease.

ix. Mixing of dataset for the improvement of performance is rare example.

x. Relationship between the Properties of Agent is not analyzed.

To meet this gap in literature the objective of the research work is designed.

i. The present work has tried to analyze the properties of Agent. It has established the relationship between different properties or semantics of Agent. Algorithmic approach is developed to increase the Agent's performance.

ii. Multi Agent system is developed and implemented using python.

iii. Multi Agent system is developed and tested with 26 different machine Learning Algorithms. Best performing algorithm is executed using Multi Agent System to achieve higher accuracy.

iv. Multi Agent System is implemented by using Backpropagation Artificial Neuron Network algorithm to achieve higher level of accuracy.

v. Agent Inheritance is implemented using Multi Agent System to attain maximum level of accuracy

Intelligent Agents can work much more accurately than human being. It can accrue certain properties like communication, autonomy, reasoning, proactivity, mobility and capability of learning etc. So the system designed , developed and implemented with Intelligent agents i.e. Multi Agent System can produce higher throughput than most of the other existing systems .So the present study infers and motivates to analyse and make a relationship between multiple Agent properties (semantics) like proactivity, learning, reasoning, mobility, etc. so that the Multi Agent System comprising of Multiple Intelligent Agents can work together to achieve better performance.

CHAPTER 4

ALGORITHMIC APPROACH FOR ANALYZING AGENT PROPERTIES TO INCREASE THE EFFICIENCY OF AGENTS IN MULTI AGENT : FORMATION OF LOGICAL SYMANTICS

We can find from the literature that there is no work done on the relationship of agent properties. Some authors have utilized the properties of agent to achieve their goal, but they have selected that very property in their paper as a solution of the basic need they wanted to fulfil .The reason for choosing that very property is not justified in those papers. Those properties are learning, reactivity, autonomy, rigidity, reasoning, rationality, negotiation etc. One more important thing is, in recent papers there is no work done by utilizing proactive agent from 2012 to 2016. So the recent work, took an attempt to calculate the efficiency in terms of Proactivity. The Relationship between properties of Agent and the algorithm to increase the efficiency is mentioned below.

I. RELATIONSHIP BETWEEN PROPERTIES OF AGENT

increase

f1: Reasoning ⟶ (Rationality, Proactivity) … 1

[Let say, increase in Reasoning causes Rationality to increase a factor of x1 and increase in Reasoning causes Proactivity to increase a factor of y1.Value of x1 and y1 can be integer or float or exponential .Based on the values of x1 and y1, we can tell whether Rationality and Proactivity has linear or exponential relationship with Reasoning]

increase

f2: Learning ⟶ (Reactivity, Autonomy) … 2

[Let say, increase in Learning causes Reactivity to increase a factor of x2 and increase in Learning causes Autonomy to increase a factor of y2. Value of x2 and y2 can be integer or float or exponential .Based on the values of x2 and y2, we can tell whether Reactivity and Autonomy has linear or exponential relationship with Learning]

increase

f3: Autonomy⟶ (Proactivity) ... 3

[Let say increase in Autonomy causes Reactivity to increase a factor of y3. Value of y2 can be integer or float or exponential .Based on the value of y2 we can tell whether Proactivity has linear or exponential relationship with Autonomy]

So we can write Proactivity = y1 * Reasoning
[From 1]

Autonomy = y2 * Learning
[From 2]

Proactivity = y3 * Autonomy
[From 3]

From 2 & 3 we get

Proactivity = Y2 * y3 * Learning … 4

So some values can be assigned as input for Learning, Reasoning and Mobility (other properties like Autonomy, Rationality etc. can be neglected and efficiency can be calculated with the help of Reasoning, Learning and Mobility because Reasoning and Learning are interrelated with Autonomy, Rationality) for each Proactive Agent and the values of y1, y2, y3 (x1, x2 if needed) according to the nature of the system. Hence, we can calculate the value of Proactivity as:

Proactivity = y1 * Reasoning ... 1

Proactivity = Y2 * y3 * Learning ... 4

So total Proactivity = (y1 * Reasoning) +

(Y2 * y3 * Learning) + Mobility
[Mobility can be given as input]

II.THE ALGORITHM TO INCREASE THE EFFICIENCY OF AGENT

Define maximum_no_of_group m

Define no_of_ maximum_ Agents_ in_a_group n

Float Reasoning[m][n];

Float Learning[m][n];

Float Mobility[m][n];

Read y1,y2,y3;

G[m];

Efficiency[m][n];

Sort_ Efficiency[m][n];

Highest_ Efficiency[m][n];

Agent[m][n];

Read Agent[m][n] //Read the value of Agents

for a particular Goal as

i) 1 in first column, if

Agent1 is needed

ii) 2 in second column, if

Agent2 is needed

….

iii) 0 if that particular Agent

is not needed

/* Coordinator Agent (CA) will start with first goal

from m number of assigned Goals */

/*Goals are arranged according to priority i.e goal1

is having highest priority*/

For I ⟵ 1 to m

Select Goal G[I] from CA

For J ⟵ 1 to n

Select Agent[I][J] for Goal G[I] //Select

first Agent from first Goal

if Agent[I][J]==0

Efficiency[I][J]= 0

else

Efficiency[I][J]=Calculate_Efficiency(Agent[I][J]) /* Efficiency matrix will hold the values of efficiency For all the Agents for all the

goals from 1 to m*/

For I ◄— 1 to m

For J ◄— 1 to n

Sort_Efficiency[I][J] = Efficiency[I][J]

//Copy the value of Efficiency

to Sort_Efficiency

Highest_Efficiency[I][J] = Efficiency[I][J]

//Copy the value of Efficiency

to Sort_Efficiency

For I ◄— 1 to m //Sort the efficiency of

Agents for Each Goal

For J ◄— 1 to n

For K ◄— J+1 to n-1

If Sort_Efficiency[I][J] < Sort_Efficiency[I][k]

Swap(&I ,& J ,& k)

/* For each group, Agents will be sorted in Descending Order .So first Agent will be the Agent with highest efficiency*/

For I ◄— 1 to m //Sort the efficiency of

Agents for Each Goal

For J ◄— 1

For K ◄— J+1 to n-1

If Sort_Efficiency[I][J] < Sort_Efficiency[I][k]

Swap(&I , &J , &k)

x=J //J is the index of highest efficiency Agent in each goal

Index[I]=x // Index[I] will store the index of highest efficiency Agent in each goal

/* Highest_efficiency is a matrix having m rows and n columns where first column stores the most efficient Agent from each goal */

/* Index array holds the column index of Agent with highest efficiency*/

/* So the Agent with highest efficiency will be called first */

Coordinator Agent can execute wait (goal no) system call when it is not necessary to start and can make signal (goal no)=True ,when want to start.

I ⟵ 1

While(true) // Implies Agent number is in between 1 to m

If signal(I)= =true

q=index[I] //q- is column index of highest efficiency Agent for goal1

No=2 // If first Agent is not free ,need to call second efficient Agent from Sort_Efficiency[I][No]

If Agent[I] [q] is free

allocate Agent[I][q]

else

Number = Calculate_next _efficient _Agent(Sort_Efficiency[I][No])

Allocate Agent[I][Number]

else

i++ //go for next goal & check

for semaphore value

/* The above mentioned procedure for calling next efficient Agent can be implemented with the help of synchronization */

/* One Agent may participate to fulfil several Goals and can be called at the same time .If a finite time(t) expires , the Agent with next higher efficiency can be called */

If Agent[I][J] is the Agent with highest efficiency

/* Check Agent[I][J] for concurrent use*/

/* use1 and use2 are two synchronized function. If an Agent is busy serving one goal (say goal2) and called from other goal (say goal4), it (goal4) has to wait .If a finite time(t) expires ,CA will start the goal with next efficient Agent by calling Calculate_next_efficient_Agent (Sort_Efficiency[I][J],I) */

Check_concurrent_use (Agent[Ii][Jj])

For J◄— Jj

For I◄— 1i

For k◄— 1 to m

If signal (Agent[I][J]) = = **signal**(Agent[K][J])

Use1 (Agent[I][J])

else

use2 (Agent[I][J])

boolean valueset=false

synchronized float use1 (Agent[I][J] **)**

if (! Valueset)

try

wait();

Catch(InterruptedException e)

Print(Agent[I][J]can not be used)

If wait (Agent[I][J]) > t

Calculate_next_efficient_Agent_Sort_Efficiency

([I][No],I)

notify all(); //Will notify other goal about the

status of present Agent

valueset=false

Notify();

Return (Proactivity(Agent[I][J]))

synchronized float use2 (Agent[I+1][J]**)**

if(valueset)

try

wait();

Catch(InterruptedException e)

Print(Agent[I+1][J]can be used)

valueset=true

Notify All();

Calculate_Efficiency (Agent [x][y])

Proactivity = (y1 * Reasoning) + (Y2 * y3 * Learning) + Mobility[x][y]

Return (Proactivity)

int Calculate_next_efficient_Agent(Sort_Efficiency[p]

[No], p)

For I ⟵ p

For J ⟵ 1 to n

If Efficiency[I][J] = = Sort_Efficiency[p][No]

Exit();

If Agent[I][J] is free

Return J;

else

Calculate_next_efficient_Agent(Sort_Efficiency[p][No+1])

/* Recursive function Calculate_next_efficient_Agent will be keep on calling the next Agent if the present highest efficiency Agent is busy and will send the column index of available highest efficiency Agent */

```
void   Swap(int *I, int *J , int *k)

        Temp  = Sort_Efficiency[I]

        Sort_Efficiency[I][J]  =  Sort_Efficiency[I][k]

        Sort_ Efficiency[I][k]  =  Temp
```

Result : For the same work, goals (In turn efficiency of agent) can be selected in different ways. The different ways can be called as choice(C1,C2,C3).This is the graph between choice and efficiency. Choice 5 i.e C5 is best.

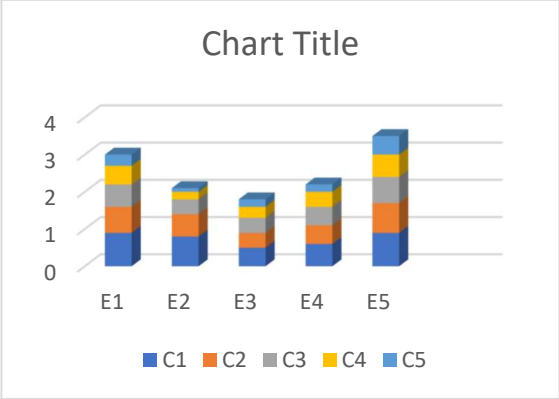


Fig 1. Graph between Choice and efficiency

CHAPTER 5

DEVELPOMENT OF MULTI AGENT SYSTEM : PREDICTION OF HEART DISEASE USING MACHINE LEARNING AND MULTI AGENT SYSTEM

The present chapter concentrates on Twenty Six different machine learning Algorithms applied on Heart Disease dataset for flawless prediction of Heart Disease .It attempts to predict heart disease using "Twenty Six " different Machine Learning Algorithms implemented by Agent. The present work also concentrates on development of Multi Agent System and trained it with the Best fit(Among Twenty Six Machine Learning Algorithms) Algorithm. Agents in the newly created Multi Agent System have Predicted the possibility of heart attack with the approximately same Accuracy. The complete work has been implemented using Python(Jupyter Notebook).

I.METHODOLOGY

1. Data Collection: In order to perform an Heart Disease prediction, the proposed method has adapted the High Dimensional dataset. At first, the Heart Disease data is created from five different datasets and used for

analysis. This Heart Disease dataset has been presented by integrating various datasets already available individually but not combined before. Hence, the integration makes it the largest Heart Disease dataset accessible for clinical research purposes.

The five datasets comprised a total of 1280 records are provided below

- Cleveland – 323
- Hungarian – 304
- Long beach VA – 280
- Stalog (heart) – 290
- Switzerland – 143

2. Preparing the Data: To make the dataset Scalable , StandardScaler() function is used. StandardScaler() function converts values from higher order to lower order. Conversion of values from higher to lower order reduces the possibility of error. Code and output of Conversion of Higher order values to Lower order is given below

```
In [12]: from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         X_train = pd.DataFrame(scaler.fit_transform(X_train), columns = X.columns)
         X_test = pd.DataFrame(scaler.transform(X_test), columns = X.columns)
```

Out[6]:

	age	sex	chest pain type	resting bp s	cholesterol	fasting blood sugar	resting ecg	max heart rate	exercise angina	oldpeak	ST slope	target
0	40	1	2	140	289	0	0	172	0	0.0	1	0
1	49	0	3	160	180	0	0	156	0	1.0	2	1
2	37	1	2	130	283	0	1	98	0	0.0	1	0
3	48	0	4	138	214	0	0	108	1	1.5	2	1
4	54	1	3	150	195	0	0	122	0	0.0	1	0

Fig 1. Original Higher order values

	age	sex	chest pain type	resting bp s	cholesterol	fasting blood sugar	resting ecg	max heart rate	exercise angina	oldpeak	ST slope
0	-0.606170	0.55561	0.830251	-0.670481	0.485989	-0.524595	-0.786977	-0.973634	-0.774856	0.965361	0.622932
1	0.688184	0.55561	0.830251	-0.137068	-2.051982	1.906232	0.361779	-0.399340	1.290563	0.158118	2.240121
2	0.903909	0.55561	-0.237023	-0.137068	0.202908	-0.524595	-0.786977	0.213241	-0.774856	0.785974	0.622932
3	-0.714033	0.55561	0.830251	0.929759	0.154101	-0.524595	-0.786977	-1.624501	1.290563	0.516892	0.622932
4	0.041007	0.55561	-1.304297	3.170094	0.710502	-0.524595	1.510535	2.089269	-0.774856	-0.828513	-0.994256

Fig 2. Lower order values

Feature Selection -Feature selection is done by ANOVA f-test.The result of feature selection is given below (both values and plotting).

```
Feature age: 59.662016
Feature sex: 75.065709
Feature chest pain type: 255.706824
Feature resting bp s: 6.793919
Feature cholesterol: 45.409481
Feature fasting blood sugar: 41.903455
Feature resting ecg: 8.855108
Feature max heart rate: 186.245664
Feature exercise angina: 232.343071
Feature oldpeak: 142.495132
Feature ST slope: 337.877510
```

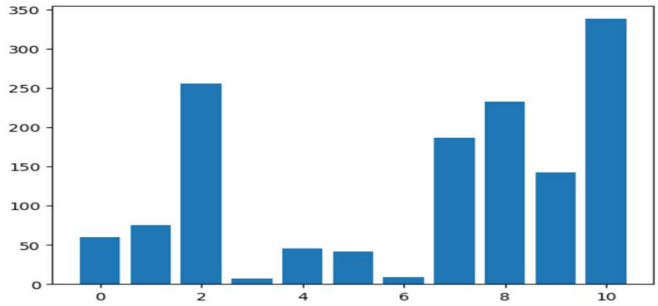


Fig 4. Feature Selection :ANOVA f-Test

3. Choosing a Model: The present work creates two different types of models. In first category, every model is separately trained with Twenty seven different algorithms by using Agent and all algorithms are separately passed in agent model .In second category ,Multi Agent model is developed and implemented the best fit algorithm among Twenty Six algorithms.

4. Training the Model: Here the dataset is splitted into two segments. One is training the model and another one is testing.70-80 percent of data can be kept for training and 20-30 percent for testing. The present work has used 67% data for training and the rest 33% of data for testing.

```
In [296]: #Declaring the train and test
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33,random_state=0)
```

Fig 5. Model Training

5. Evaluating the Model:

This phase checks the performance of the model which is done by testing the performance of the model on previously unseen data.The hybrid data was already splitted and the Model was tested with testing data(33%).Twenty Six different Machine Learning algorithms got tested and predicted by using Agent and predicts Accuracy,BalanceAccuracy,ROC-AUC,F1-Score.Those scores of all the algorithm is given below.

```
In [174]: import pandas as pd
db={"Model":"DummyClassifier"  ,"Accuracy":accuracy,"Balance Accuracy":bs,"ROC AUC":rc,"F1-Score":score_dummy,'exec_time':[time_
#result_df=result_df.append(db,ignore_index=True)
result_df=pd.concat([result_df,pd.DataFrame(db)],ignore_index=True)
result_df
```

Out[174]:

	Model	Accuracy	Balance Accuracy	ROC AUC	F1-Score	exec_time
0	Logistic Regr.	0.853428	0.853571	0.853571	0.853428	0.015957
1	SVC	0.924350	0.922705	0.922705	0.924350	0.030921
2	RF	0.933806	0.932747	0.932747	0.933806	0.254318
3	DT	0.860520	0.861912	0.861912	0.860520	0.009971
4	GNB	0.855792	0.856225	0.856225	0.855792	0.002992
5	knn	0.867612	0.867017	0.867017	0.867612	0.010005
6	sgdc	0.770686	0.772694	0.772694	0.770686	0.012001
7	perceptron	0.735225	0.731941	0.731941	0.735225	0.001997
8	lsvc	0.853428	0.853571	0.853571	0.853428	0.040858
9	Nusvc	0.862884	0.861330	0.861330	0.862884	0.035905
10	QuadraticDiscriminantAnalysis	0.843972	0.844290	0.844290	0.843972	0.023934
11	NearestCentroid	0.853428	0.853952	0.853952	0.850000	0.001994
12	CalibratedClassifierCV	0.853428	0.853762	0.853762	0.853428	0.026929
13	BernoulliNB	0.848700	0.848645	0.848645	0.848700	0.004986
14	PassiveAggressiveClassifier	0.782506	0.782725	0.782725	0.782506	0.002993
15	RidgeClassifierCV	0.851064	0.851299	0.851299	0.851064	0.591657
16	RidgeClassifier	0.851064	0.851299	0.851299	0.851064	0.075796
17	LinearDiscriminantAnalysis	0.851064	0.851299	0.851299	0.851064	0.005987
18	AdaBoostClassifier	0.851064	0.851108	0.851108	0.851064	0.233417
19	BaggingClassifier	0.900709	0.899597	0.899597	0.900709	0.328122
20	ExtraTreesClassifier	0.936170	0.935401	0.935401	0.936170	0.201459
21	LGBMClassifier	0.921986	0.922335	0.922335	0.921986	0.423145
22	XGBClassifier	0.931442	0.931236	0.931236	0.931442	0.251867
23	LabelSpreading	0.912530	0.912483	0.912483	0.912530	0.042886
24	LabelPropagation	0.912530	0.912483	0.912483	0.912530	0.028922
25	DummyClassifier	0.520095	0.500000	0.500000	0.520095	0.000999

Fig 6. Accuracy,Balance Accuracy,ROC-AUC,F1-Score and Execution time are calculated for all the algorithms used to predict Heart disease.

Graphical Representation of Accuracy of different Algorithms are given below

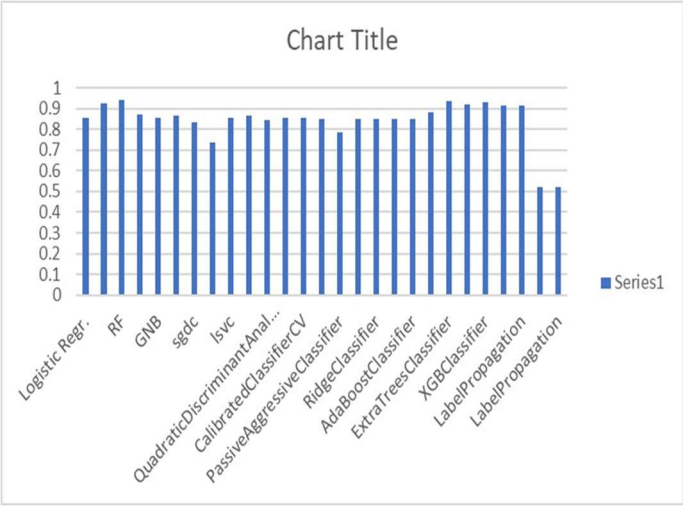


Fig 7. Graphical Representation of Accuracy of different Algorithms are given below

6.Making Predictions: Multiple Agents have been created and trained with RandomForestClassifier() as it is proved as the best accurate algorithm for the present dataset. Newly trained Multi Agent System is able to detect the possibility of heart disease with almost the same accuracy.

```
class Agent1:
    def __init__(self):
        self.model = RandomForestClassifier(n_estimators=20,criterion='gini')

    def train(self, X, y):
        self.model.fit(X, y)
    def predict(self, X):
        return self.model.predict(X)
# Define Agent 2
class Agent2:
    def __init__(self):
        self.model = RandomForestClassifier()
    def train(self, X, y):
        self.model.fit(X, y)
    def predict(self, X):
        return self.model.predict(X)
# Train and predict using agents
agent1 = Agent1()
agent2 = Agent2()
# Train Agent 1
agent1.train(X_train, y_train)
# Get predictions from Agent 1
predictions_agent1 = agent1.predict(X_test)
# Train Agent 2
agent2.train(X_train, y_train)
# Get predictions from Agent 2
predictions_agent2 = agent2.predict(X_test)
print("Agent1 accuracy=",accuracy_score(y_test, predictions_agent1))
print("Agent2 accuracy=",accuracy_score(y_test, predictions_agent2))
```

```
Agent1 accuracy= 0.96875
Agent2 accuracy= 0.96484375
```

By improving accuracy ……..

In [1]:
```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import pandas as pd

# Assume you have a dataset with features and labels
# Replace 'your_dataset.csv' with your actual dataset
# Replace 'target_column' with the column containing the target variable
data = pd.read_csv('1681058250160_1680773890113_Data Set.csv')
X = data.iloc[:,:-1].values
y = data.iloc[:,-1].values

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Define Agent 1

class Agent1:
    def __init__(self):
        self.model = RandomForestClassifier(n_estimators=20,criterion='gini')

    def train(self, X, y):
        self.model.fit(X, y)
    def predict(self, X):
        return self.model.predict(X)
# Define Agent 2
class Agent2:
    def __init__(self):
        self.model = RandomForestClassifier()
    def train(self, X, y):
        self.model.fit(X, y)
    def predict(self, X):
        return self.model.predict(X)
# Train and predict using agents
agent1 = Agent1()
agent2 = Agent2()
# Train Agent 1
agent1.train(X_train, y_train)
# Get predictions from Agent 1
predictions_agent1 = agent1.predict(X_test)
# Train Agent 2
agent2.train(X_train, y_train)
# Get predictions from Agent 2
predictions_agent2 = agent2.predict(X_test)
print("Agent1 accuracy=",accuracy_score(y_test, predictions_agent1))
print("Agent2 accuracy=",accuracy_score(y_test, predictions_agent2))
```

```
Agent1 accuracy= 0.953125
Agent2 accuracy= 0.9609375
```

By improving accuracy ……..

```
class Agent1:
    def __init__(self):
        self.model = RandomForestClassifier(n_estimators=20,criterion='gini')

    def train(self, X, y):
        self.model.fit(X, y)
    def predict(self, X):
        return self.model.predict(X)
# Define Agent 2
class Agent2:
    def __init__(self):
        self.model = RandomForestClassifier()
    def train(self, X, y):
        self.model.fit(X, y)
    def predict(self, X):
        return self.model.predict(X)
# Train and predict using agents
agent1 = Agent1()
agent2 = Agent2()
# Train Agent 1
agent1.train(X_train, y_train)
# Get predictions from Agent 1
predictions_agent1 = agent1.predict(X_test)
# Train Agent 2
agent2.train(X_train, y_train)
# Get predictions from Agent 2
predictions_agent2 = agent2.predict(X_test)
print("Agent1 accuracy=",accuracy_score(y_test, predictions_agent1))
print("Agent2 accuracy=",accuracy_score(y_test, predictions_agent2))
```

```
Agent1 accuracy= 0.96875
Agent2 accuracy= 0.96484375
```

Fig 8. RandomForestClassifier() is implemented to train and test Multiple Agents :Produced accuracy approximately same

II. RESULT

The present work has used Twenty Six different Machine Learning Algorithms. for prediction of Heart disease by using Hybrid dataset(for Heart disease) Accuracy, Balance Accuracy,ROC-AUC,F1-Score and Execution time is calculated and compared for all Twenty Six Algorithms. RandomForestClassifier() have been proved as the best for accurate prediction of heart attack.

CHAPTER 6

TESTING AND IMPLEMENTATION OF MULTI AGENT SYSTEM : PREDICTION OF HEART DISEASE USING ARTIFICIAL NEURAL NETWORK

Earlier chapter has focused on development of Multi Agent System. Multi Agent System is applied to implement Machine Learning Algorithms. The recent chapter has predicted Heart Disease using Back Propagation Artificial Neural Network implemented by Multi Agent System

I.METHODOLOGY

Five different datasets have been merged together to get more accurate prediction of Heart disease. How each and every individual parameter of the dataset is responsible for Heart disease is altogether represented graphically.

```
In [10]: import matplotlib.pyplot as plt
         plt.subplot(11,1,1)
         plt.plot(range(0,df.shape[0]),df.iloc[:,0],label="age")
         plt.legend()
```

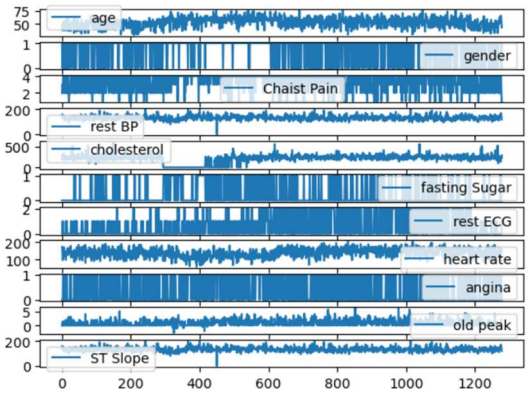


Fig 1. How each and every individual parameter of dataset is responsible for Heart attack is represented graphically

```
In [11]: df.hist(figsize =(12, 12))
         plt.show()
```

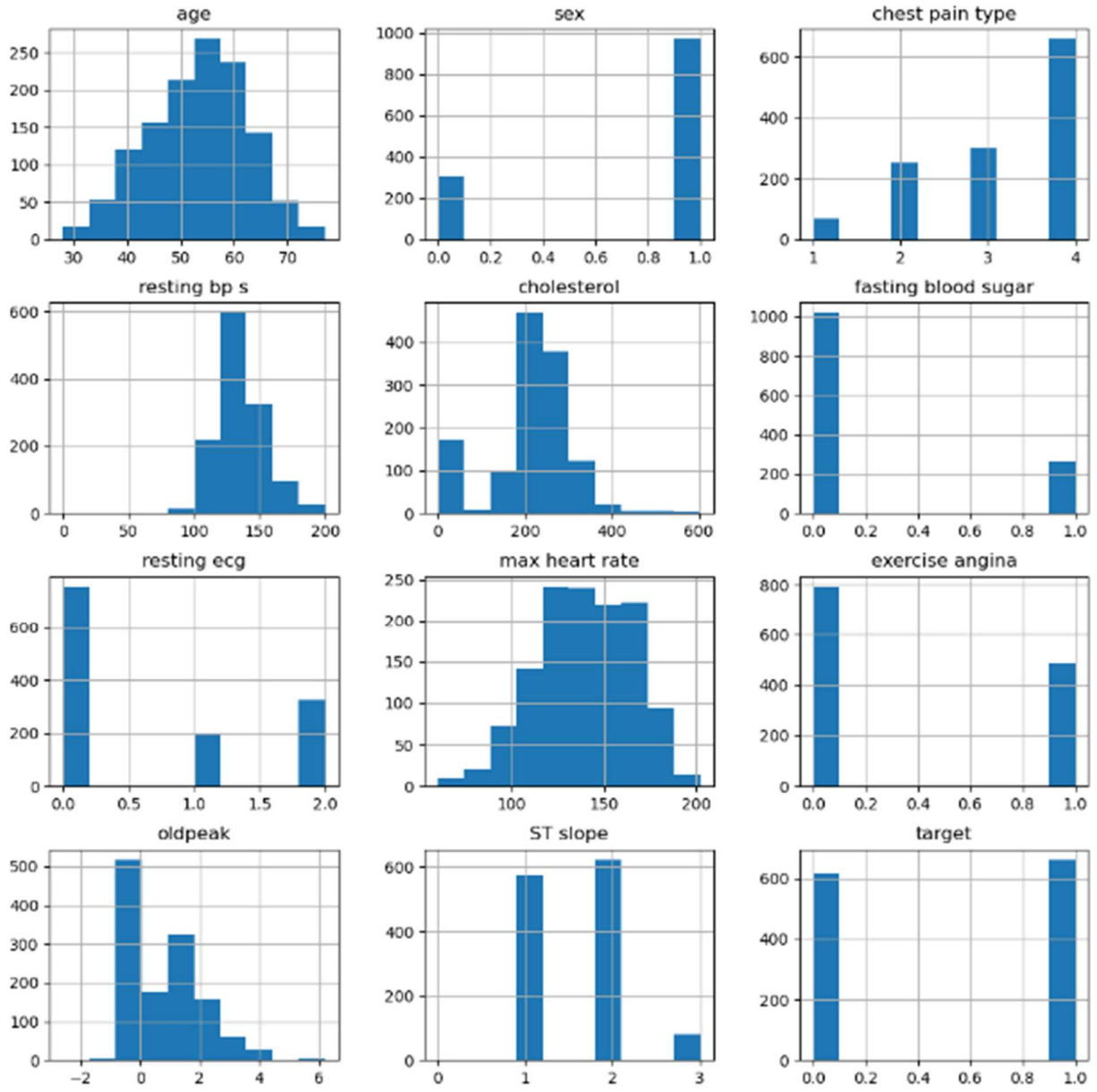


Fig 2. Bar graph for individual features to provide a comprehensive view of the empirical outcomes.

Correlation matrix-A statistical method called correlation matrix is used to establish the relationship between two variables in this Hybrid dataset. This matrix is a table in which every cell contains a correlation coefficient , where 1 is considered as a strong relationship between variables, 0 as a neutral relationship and -1 as a not strong relationship. From the diagram below, it is clear that Target is dependent on ST slope because Target-ST slope value is positive. Target is less dependent on Cholesterol because Target-cholesterol value is negative.

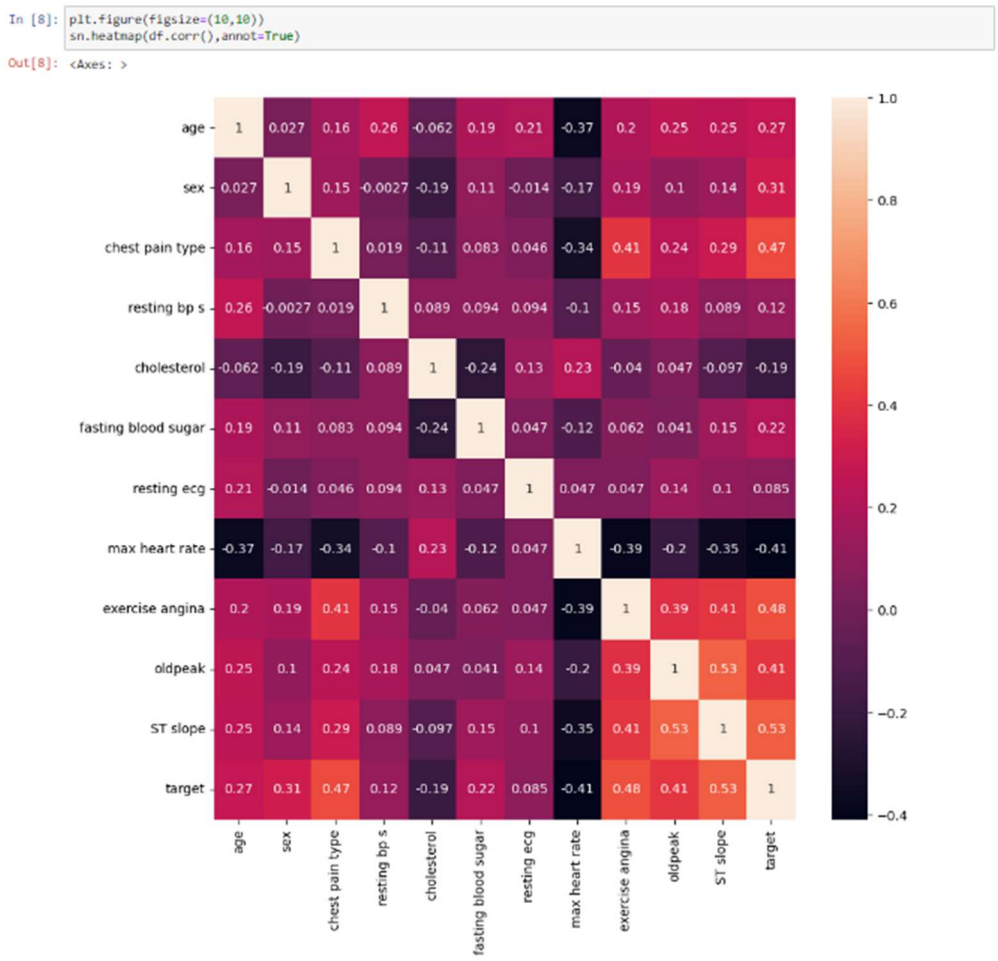


Fig 3. Correlation Matrix

II.PREDICTION MAKING

The most innovative invention of this work is creation of Multi Agent based Artificial Neural Network Model. Multiple layer is created i.e dense layer ,dropout layer etc. In previous Multi Agent Model ,Model was implemented with single layer. The present work has implemented (using Jupyter Notebook) ANN Multi Agent model with multiple layers with inheritance to improve accuracy. Neural Network model is trend with Agent (say Agent1) and then Agent 1 is passed to Agent 2. Agent 2 has a copy of Agent 1 and extra layers are also added to Agent 2. Neural Network model trend with Agent 1 and Agent 2 (inherited from Agent 1) , having a copy of Agent 1 with extra added layer, produced accuracy of 99.7%.

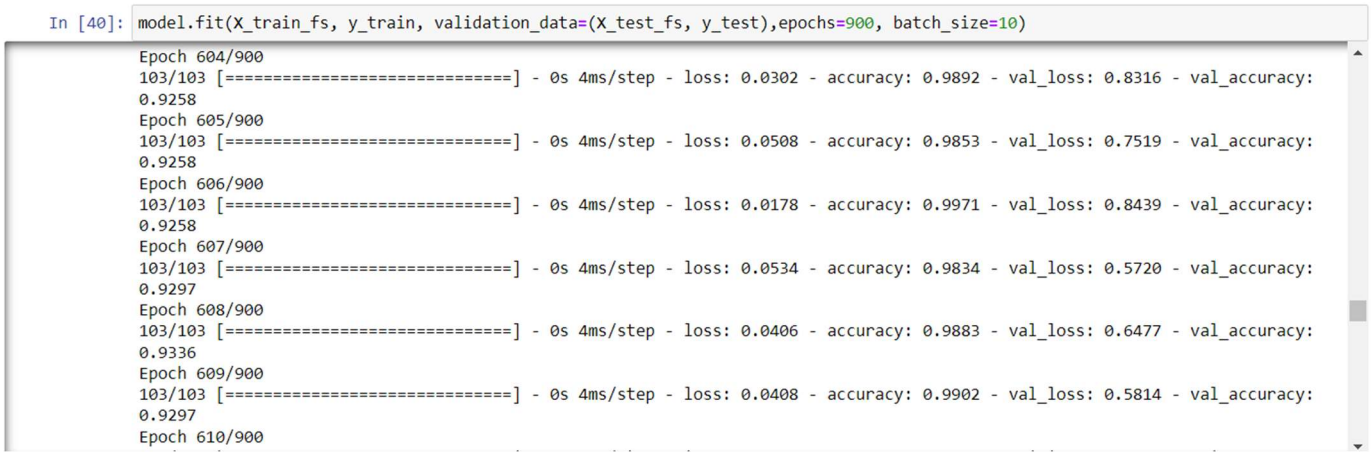

In [40]: model.fit(X_train_fs, y_train, validation_data=(X_test_fs, y_test),epochs=900, batch_size=10)
Epoch 604/900
103/103 [==============================] - 0s 4ms/step - loss: 0.0302 - accuracy: 0.9892 - val_loss: 0.8316 - val_accuracy:
0.9258
Epoch 605/900
103/103 [==============================] - 0s 4ms/step - loss: 0.0508 - accuracy: 0.9853 - val_loss: 0.7519 - val_accuracy:
0.9258
Epoch 606/900
103/103 [==============================] - 0s 4ms/step - loss: 0.0178 - accuracy: 0.9971 - val_loss: 0.8439 - val_accuracy:
0.9258
Epoch 607/900
103/103 [==============================] - 0s 4ms/step - loss: 0.0534 - accuracy: 0.9834 - val_loss: 0.5720 - val_accuracy:
0.9297
Epoch 608/900
103/103 [==============================] - 0s 4ms/step - loss: 0.0406 - accuracy: 0.9883 - val_loss: 0.6477 - val_accuracy:
0.9336
Epoch 609/900
103/103 [==============================] - 0s 4ms/step - loss: 0.0408 - accuracy: 0.9902 - val_loss: 0.5814 - val_accuracy:
0.9297
Epoch 610/900

Fig 4. Prediction of heart disease using ANN and Multi Agent System with accuracy of 99.7 %.

III. RESULT

Back Propagation Neural network Algorithm is implemented with Multi Agent System and Multi level inheritance and predicted Heart disease with 99.7% accuracy.

CHAPTER 7

APPLICATION OF MULTIAGENT SYSTEM IN HEALTHCARE

I. PROPOSED SYSTEM

Present work has created A Multi Agent System which symbolizes a population of Five Hundred people where one Agent is initially infected. Objective of this work is to design a Multi Agent based simulation model[6][7] to show the spread of coronavirus based on Euclidean distance between every pair of Agents and their inbuilt resistance and recovering the affected Agents from infection. Simulation of Multi Agent System [8][9] is shown with the help of image viewer in place of video, as video consumes less space in Memory. Multi Agent [10]Simulation is implemented using Jupyter notebook.

II. SPREAD OF CORONAVIRUS

1. Creation of population with the Intelligent Agents

2. Creation of Agent class

3. Individual initially infected

4. Spreading of Infection from one Agent to another based on Euclidean distance and self-resistance

3. Recovered from the stage of infection

The way of transformation between all three stages can ultimately decide the actual balance of infectious and recovered population.

1. Creation of population with the Intelligent Agents : The present work has simulated the spread of Coronavirus with the help of a Multi Agent System having Five Hundred Agents. Every individual in the population is represented as an Agent .

2. Creation of Agent Class : Intelligent Agent of Multi Agent System is created with the help of Agent class. Five Hundred object of Agent class is created initially . Two instance variables x, y are used to initialize and track the position of all the agents.

3. Individual initially infected : Infected counter = 1 means positive case. Every time the infected counter is incremented by 50. In every step 10 are getting infected , 10 instances are changing color from green to red approximately . Step size is taken as 5.

4. Spreading of Infection from one Agent to another based on Euclidean distance and self resistance : X and Y axis value of individual Agents are continuously getting changed randomly to change their position And in turns incorporating mobility. With the movement of Agent iteratively , the Euclidean distance between every two agent changes. After every iteration the number of newly affected Agents need to

be measured. One Agent will be initially made as infected and based on the Euclidean distance the nearby agents start getting affected. The number of Agent to be affected depends on the value of Euclidean distance and the value of resistance. All the agents are marked as green. As and when they are getting affected their color is made to change to red. Every movement is recorded in a frame. Five Hundred frames are generated for Five Hundred iterations (movements).

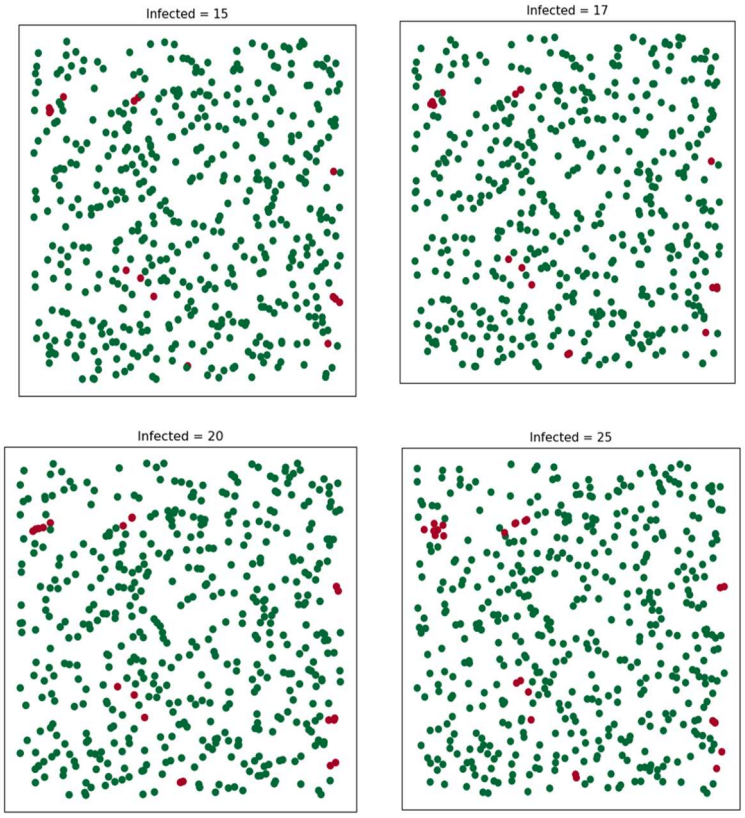


Fig 1. Spreading of Coronavirus based on Euclidean distance and self-resistance (First four frames)

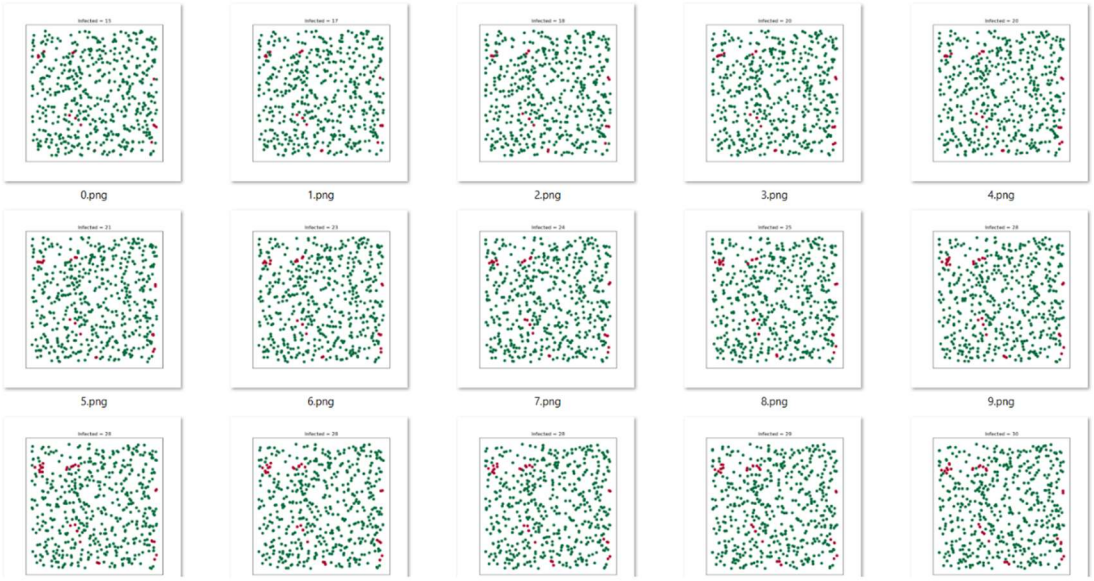
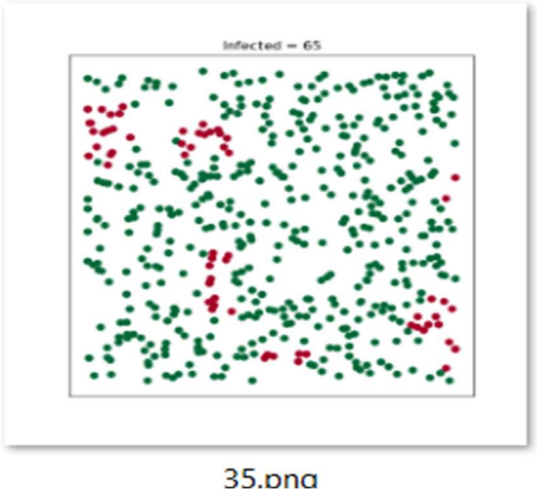
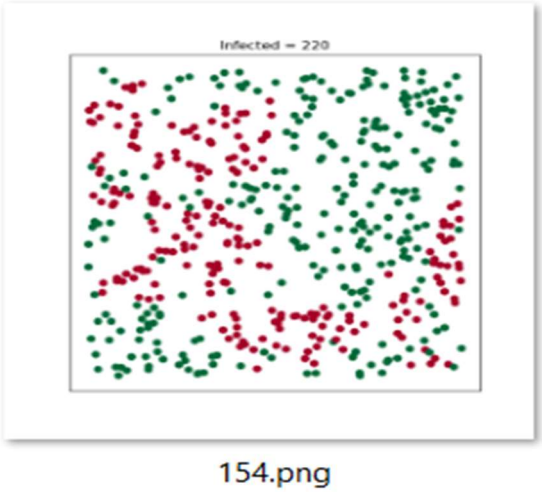


Fig 2. Simulation diagram of spread of coronavirus up to fifteenth iteration

5. Recovered from the stage of infection: Infect function will be executed when one non-infected Agent come in contact with an infected agent. Infect function generates a random number for every interaction between Agents. If the random number is greater than the Agent resistance then the Agent will move to infectious state. The present work has experimented with agents having different resistance value, as in our society, people wearing a mask may come near affected people but do not get affected. So, instead of having less Euclidean distance also Agent may not turn into Infectious state. When the Agent is infected, its infected counter is set to some value and every time step the counter, start ticks down. After counter becomes zero Agent gets healthy again. Setting high resistance value after the Agent is recovered can ensure the chance of reinfection.





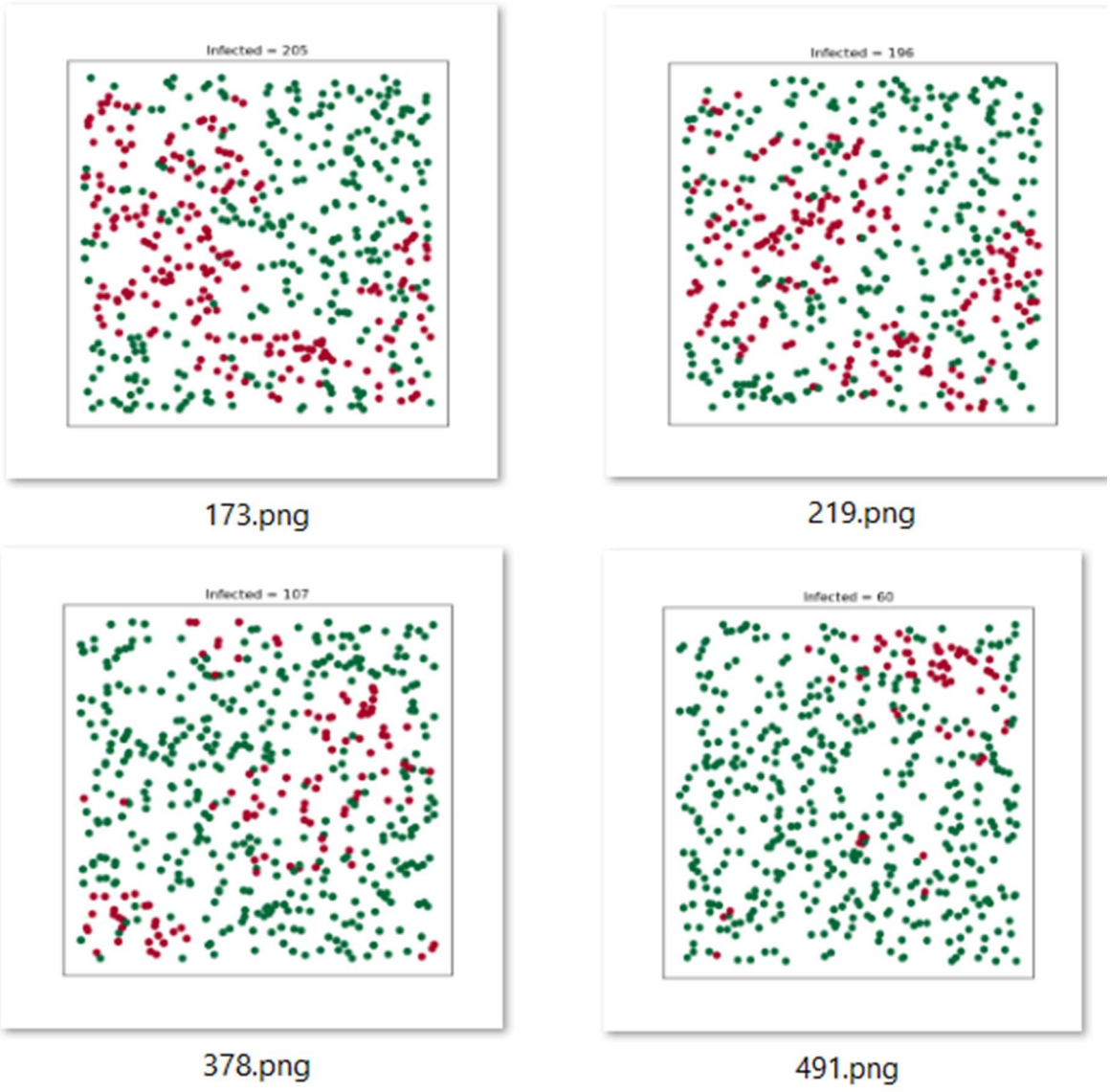


Fig 3. Frame 35 - Infected 65, Frame 154 - Infected 220, Frame 173- Infected 205,Frame 219 - Infected 106, Frame 378 - Infected 107, Frame 491- Infected 60

In Frame number 35, number of infected are 65 where as in Frame 491, the value of infected are 60 which proves that there are huge number of recoveries from coronavirus.

III.RESULT

Progression of Frame symbolizes spread of coronavirus. Frame number 2 is next step of progression of coronavirus after frame number 1.Figure 1 shows first four Frames which implies the simulated steps of the progression of coronavirus. It is clear from first four figure that number of infected increases from first to fourth frame. From the figure 3 it is observed that in Frame 35, 154, 173 number of Infected are 65, 220, 205 and again in Frame 219,378,491 number of infected are 106,107, 60 respectively. From the above

result it is obvious that if Frame number increases spread of Coronavirus does not increase proportionately, it decreases after a certain stage but not equivalently. In Frame number 35, number of infected are 65 where as in Frame 491, the value of infected are 60 which proves that there are huge number of recoveries from coronavirus.

Fig 4. First 50 Frames

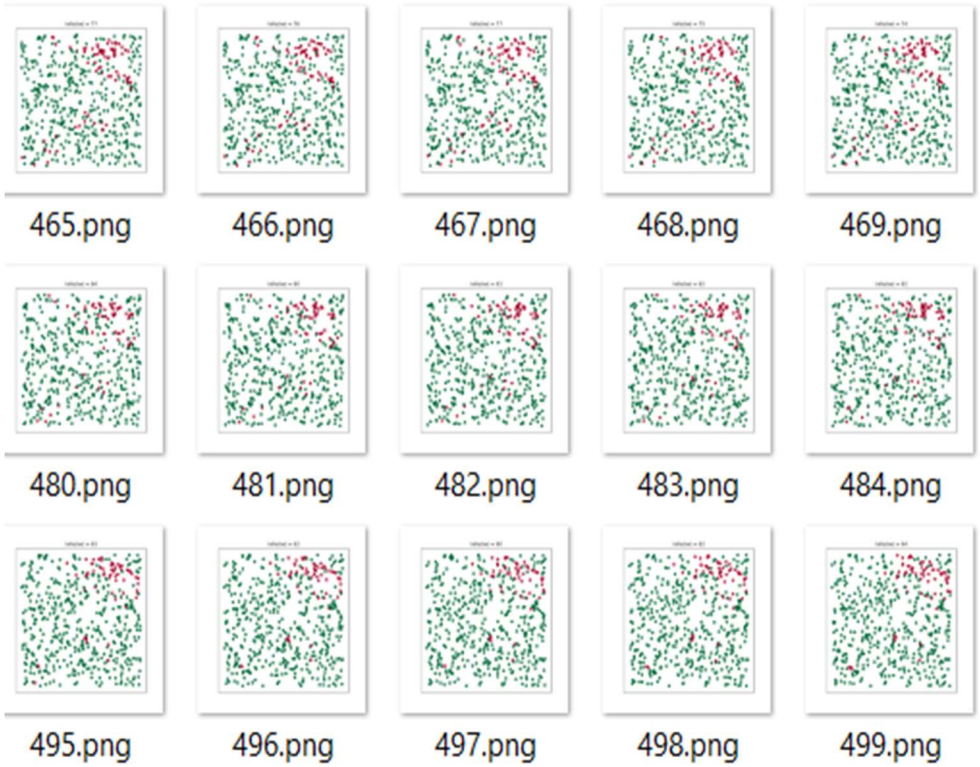


Fig 5. Simulation number 465 to 499

Simulation of Spread of coronavirus is shown above. Infected vs Non-Infected graph is presented below based on one thousand simulation.

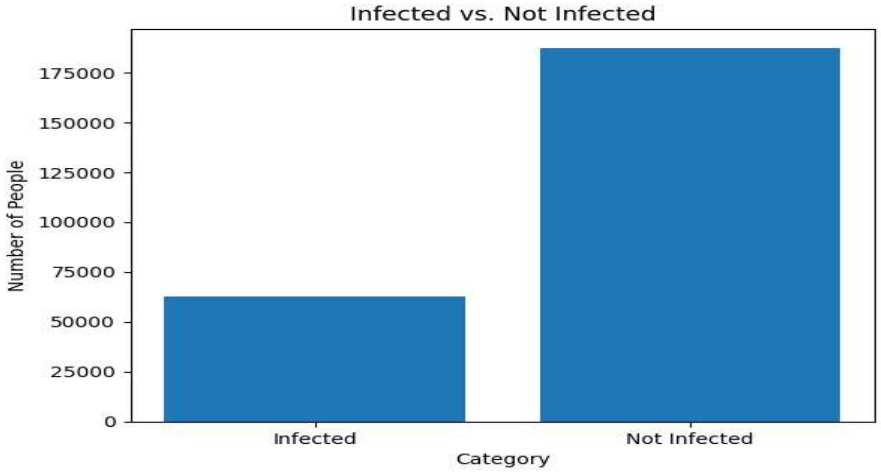


Fig 6. Infected Vs Not Infected graph is generated based on One thousand simulation.

CHAPTER 8

CONCLUSION

Objective of the present work is to analyse semantics of Agents and develop a Multi Agent System. Machine Learning Algorithms have been used to implement Multi Agent System. The work concentrates on Twenty Six different machine learning Algorithms applied on Heart Disease dataset for flawless prediction of Heart Disease, its Accuracy, Balance Accuracy, ROC-AUC,F1-Score ,Execution time etc. The present work attempts to predict heart disease using “Twenty Six " different Machine Learning Algorithms implemented by Agent and predicted Heart Disease with 93.8% accuracy. The paper also concentrates on development of the Multi Agent System and training it with the Best fit (Among Twenty-Six Machine Learning Algorithms) Algorithm. Agents in the newly created Multi Agent System have predicted the possibility of heart disease with an Accuracy of 96.8% approximately. The complete work has been implemented using Python (Jupyter Notebook). Implementation of Multi Agent System with Machine Learning Algorithm is the novelty of the present research work which can explore a new era in the Field of Artificial Intelligence. By systematically comparing the performance of these algorithms, it is identified that most effective models are required to refine and optimize prediction accuracy. So, further the final outcome of this research is the development of a hybrid Artificial Neural Network (ANN)-based algorithm, designed and implemented using Python in a Jupyter Notebook environment. This hybrid ANN algorithm, leveraging the collaborative capabilities of intelligent agents, achieved a remarkable accuracy rate of 99.7%. The results underscore the transformative potential of multi-agent systems in medical diagnostics, paving the way for more accurate and reliable heart disease prediction models. This innovative research work has implemented Multi Agent System for eradication of Pandemic.

REFERENCES

[1] Katia P.Sycara.1998. Multi Agent Systems, AI Magazine 19(2) pp 79-92,1998.

[2] Nicholas R. Jennings, Katia P. Sycara and Michael Wooldridge. A Road map of Agent Research and Development. Autonomms Agents and Multi-Agent System, K1uwer Academe Publishers, Boston Manufactured in The Nether1ands, 1,275-306,1998.

[3] Terry R. Payne, Massimo Paolucci, Rahul Singh, and Katia Sycara. Communicating Agents in Open multi agent systems. First GSFC/JPL Workshop on Radical Agent Concepts (WRAC), McLean, VA,USA, 16 - 18 Jan 2002. , 365-371.

[4] Regina Sousa, Diana Ferreira, Antonio Abelha, and Jose Machado, Step Towards Monitoring Intelligent Agents in Healthcare Information Systems

[5] Maroua Ahmid, Okba Kazar, Saber Benharzallah,,Laid Kahloul, Abdelhak Merizig, An Intelligent and Secure Health Monitoring System Based on Agent, DOI: 10.1109/ICIoT48696.2020.9089602, INSPEC Accession Number: 19594461,IEEE

[6] Maroua Ahmid, Okba Kazar, A Cloud-IoT Health Monitoring System Based on Smart Agent for Cardiovascular Patients, DOI: 10.1109/ICIT52682.2021.9491113, INSPEC Accession Number: 20945454

[7] Normadiah mahiddin, Zulaiha Ali Othman,Azuraliza Abu Bakar, Nur Arzuar, Abdul Rahim. An Interrelated Decision-Making Model for an Intelligent Decision Support System in Healthcare, DOI: 10.1109/ACCESS.2022.3160725, date of publication March 21, 2022, date of current version March 25, 2022.

[8] Islam Daoud Suliman,D. Vasumathi,"Prediction of Heart Disease Using Machine Learning Algorithms",IEEE,23 November,2023,10.1109/ICCCNT56998.2023.10308096,https://ieeexplore.ieee.org/xpl/conhome/10306338/proceeding, doi.org/10.1109/ICCCNT56998.2023.10308096

[9] Chintan Bhatt,ParthPatel,Tarang Ghetia "Effective Heart Disease Prediction Using Machine LearningTechniques" https://doi.org/10.3390/a16020088,https://www.mdpi.com/1999-4893/16/2/88

[10] Niloy Biswas,Md Mamun Ali,Md Abdur Rahaman,Minhajul Islam,Md. Rajib Mia,"Machine Learning-Based Model to Predict Heart Disease in Early Stage Employing Different Feature Selection Techniques",BioMed Research ,Volume 2023, ArticleID-6864343 ,Int.https://doi.org/10.1155/2023/6864343

[11] Dhanunjaya Rao,B. Amani Rama Padma Priya,B. Venkata Sai Sowmyal,T. Chandrasekhar,"Machine Learning Algorithms For Prediction Of Heart Disease",International Research Journal of Modernization in Engineering Technology and Science,e-ISSN: 2582-5208,DOI : https://www.doi.org/10.56726/IRJMETS35616

[12] Nadikatla Chandrasekhar et al.,"Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization",2023, https://doi.org/10.3390/pr11041210

[13] Huating Sun, Jianan Pan,"Heart Disease Prediction Using Machine Learning Algorithms with Self-Measurable Physical Condition Indicators",Journal of Data Analysis and Information Processing, 2023,ISSN Online: 2327-7203,doi.org/10.4236/jdaip.2023.111001

[14] Md. Imam Hossain,Mehadi Hassan Maruf,Md Asikur Rahman Khan,"Heart disease prediction using distinct artificial intelligence techniques: performance analysis and comparison",Springer Link,12 June 2023,https://doi.org/10.1007/s42044-023-00148-7

[15] G. Harinadha Babu,Gunda Jayasree,Chattu Ashika,Vajja Ahalya,Katta Asha Niroopa.,"Heart Disease Prediction System Using Random Forest Technique"https://doi.org/10.22214/ijraset.2023.48764

[16] Neha Nandal,Rohit Tanwar,"Machine learning-based heart attack prediction: A symptomatic heart attack prediction method and exploratory analysis",This article is included in the Computational Modelling and Numerical Aspects in Engineering collection,https://doi.org/f1000research.com/articles/11-1126

[17] Janaraniani N,Lipika Goel,Rohit Tanwar,"Heart Attack Prediction using Machine Learning",IEEE,21-23 September 2022,https://doi.org/10.1109/ICIRCA54612.2022.9985736

[18] Chaimaa Boukhatem,Heba Yahia youssef,Ali Bou Nassif, "Heart Disease Prediction Using Machine Learning",IEEE, 18 March ,2022,https://doi.org/10.1109/ASET53988.2022.9734880

[19] P. Chinnasamy,,"Machine learning based cardiovascular disease prediction",IEEE,21-23 September 2022,https://doi.org/10.1109/ICIRCA54612.2022.9985736

[20] Sumaira Ahmed,Salahuddin Shaikh,Farwa Ikram,"Prediction of Cardiovascular Disease on Self-Augmented Datasets of Heart Patients Using Multiple Machine Learning Models",Volume 2022 | Article ID 3730303 https://doi.org/10.1155/2022/3730303

[21] Victor Chang,Vallabhanent Rupa Bhavani,Ariel Qianwen Xu,,"An artificial intelligence model for heart disease detection using machine learning algorithms",Science Direct,Volume,2,Nov 2022,https://doi.org/ 10.1016/j.health.2022.100016

[22] Mihir Patel,Rahul Patange,Chaitanya Patil,"Predicting Heart Disease Using Machine Learning Algorithms",IRJET,Volume: 09 Issue: 04 Apr 2022,e-ISSN: 2395-0056,p-ISSN: 2395-0072

[23] Jian Yang,Jinhan Guan.,"A Heart Disease Prediction Model Based on Feature Optimization and Smote-Xgboost Algorithm"https://doi.org/10.3390/info13100475

[24] Harshit Jindal,Sarthak Agaewal,Rishabh Khera,Rachna Jain,Preeti Nagrath,"Heart disease prediction using machine learning algorithms" IOP Conf. Series: Materials Science and Engineering,https://doi.10.1088/1757-899X/1022/1/012072

[25] Apurv Garg.Bhartendu Sharma,Rijwan Khan,"Heart disease prediction using machine learning techniques",IOP Conf. Series: Materials Science and Engineering,doi.org/10.1088/1757-899X/1022/1/012046

[26] Elwahsh H,"A new smart healthcare framework for real-time heart disease detection based on deep and machine learning",doi.org/10.1088/1757-899X/1022/1/012046

[27] Qin Fan, Qun Li, Youliang Chen & Jianbo Tang, "Modeling COVID-19 spread using multi-agent simulation with small-world network approach",Published: 02 March 2024, BMC Public Health volume 24, Article number: 672 (2024)

[28] Bin Chen, Runkang Guo , Zhengqiu Zhu, Chuan Ai, and Xiaogang Qiu, "Simulation of COVID-19 Outbreak in Nanjing Lukou Airport Based on Complex Dynamical Networks", DOI: 10.23919 /CSMS.2023.0001, ISSN 2096-9929 06/06 pp 71−82 Volume 3, Number 1, March 2023, COMPLEX SYSTEM MODELING AND SIMULATION.

[29] Jamlech Iram N. Gojo Cruz; Maria Art Antonette D. Clariño, "An Agent-based Modeling and Simulation Tool for the Exploration of the Effects of Masks and Vaccines on COVID-19 Spread", https://doi.org/10.1109/iCORE58172.2022. 00031,28 March 2023,IEEE.

[30] Yangyi Zhang, Sui Tang & Guo Yu, "An interpretable hybrid predictive model of COVID-19 cases using autoregressive model and LSTM", 13, Article number: 6708 ,2023.

[31] Pritpal Singh,Yo-Ping Huang , "An Ambiguous Edge Detection Method for Computed Tomography Scans of Coronavirus Disease 2019 Cases",https://doi.org/10.1109/TSMC/2023.3307393,12 September 2023,IEEE.

[32] saket kumar,rajkumar, "viral effect, challenges, and forecasting of covid-19 situation in india using an arma model", https://doi.org/10.1109 /TCSS.2021.3059773,30 July 2021,IEEE.

[33] M. Rohini,K. R. Naveena,G. Jothipriya,S. Kameshwaran,M. Jagadeeswari,A Comparative Approach To Predict Corona Virus Using Machine Learning,https://doi.org/10.1109/ICAIS50930.2021.9395827,12 April 2021,IEEE.

[34] Rajaa Aboulaich,Khalid Bensaid, Salma Chabbar,Jaafar El Karkri,"Mathematical modeling and multi-agents approach for the evolution of the Coronavirus pandemic", DOI: 10.1109/ICTMOD49425.2020.9380596., Nov 24, 2020,IEEE.

[35] Cliff C . Kerr, Robyn M. Stuart, Dina Mistry, Romesh G Abeysuriya,"Covasim: an agent-based m odel of COVID-19 dynamics and interventions", doi: https://doi.org/10.1101/2020.05.10.20097469, May 15, 2020.,NIH.

[36] Olga Krivorotko, Mariia Sosnovskaia, Ivan Vashchenko, Cliff Kerr ,Daniel Lesnic,“Agent-based modeling of COVID-19 outbreaks for New York state and UK: Parameter identification algorithm”, https://doi.org/10.1016/ j.idm.2021.11.004 , 23 June 2021.